Prototype-Enhanced Explainable Recommendation with Synthetic Reviews

Wout Kooijman*
University of Amsterdam
Amsterdam, the Netherlands
wout.kooijman@student.uva.nl

Kai Liang*
University of Amsterdam
Amsterdam, the Netherlands
kai.liang@student.uva.nl

Weitao Luo*
University of Amsterdam
Amsterdam, the Netherlands
weitao.luo@student.uva.nl

Erik Stammes*
University of Amsterdam
Amsterdam, the Netherlands
erik.stammes@student.uva.nl

Ozzy Ülger*
University of Amsterdam
Amsterdam, the Netherlands
ozzy.ulger@student.uva.nl

Victoria Foing*
University of Amsterdam
Amsterdam, the Netherlands
victoria.foing@student.uva.nl

ABSTRACT

Recommender systems are putting more emphasis on explainability as it can significantly enhance user trust in the system [30]. A new approach of explainable recommender systems is to generate synthetic reviews to supplement rating predictions. Multimodal Review Generation (MRG) is a multi-task model that performs both rating prediction and review text generation and demonstrates that unifying these two tasks could lead to more accurate ratings and relevant synthetic reviews [25]. Though MRG is a state-of-the-art model, we believe it can achieve better performance on both tasks with a prototype editor. Hence, we propose a new model, MRG + Prototype Editor + Attention (MRG-PEA), which uses a prototype editor with an attention mechanism to pass additional embeddings to the review generator. In this paper, we perform a survey of existing methods for the explainable recommendation, comparing them to the new model design. Following that, we conduct experiments using the Yelp dataset where we compare the performance of our extended model, MRG-PEA, to that of MRG on the tasks of rating prediction and review text generation. Results indicate that the MRG-PEA outperforms the MRG model on both tasks, highlighting the potential of prototype editors with attention in the explainable recommender systems.

KEYWORDS

rating prediction, review generation, explainable recommendation, prototype editing, attention, multi-task learning

1 INTRODUCTION

Explainability is becoming a powerful addition to recommender systems. When users receive an explanation for the recommendation, they are more likely to trust the recommendation and be satisfied with the system [30]. This paper seeks to advance the capabilities of explainable recommender systems by investigating the strengths and weaknesses of existing methods and proposing a solution that extends the Multimodal Review Generation (MRG) model [25]. The goal of the extended model is to improve performance on the following two tasks: given a user and item pair, the model should (1) predict the rating that the user would give the item, and (2) generate a personalized review of the item that is relevant to the user.

To begin, a literature review of the field of explainable recommendations is carried out, comparing existing methods to our proposed solution. The range of existing solutions for explaining recommendations is vast due to the fact that there are many combinations of explainable recommendation models and types of explanations [30]. As a result, we focus on solutions using deep learning models and textual explanations. The solutions investigated are divided into methods that provide explanations based on the content of the input reviews and methods that generate synthetic reviews. Emphasis is placed on the latter methods as they explain recommendations using natural language and address the issue of sparsity in existing reviews.

The literature review begins by examining content-based recommender systems such as Deep Cooperative Neural Networks (DeepCoNN) [32], Neural Attentional Regression model with Review-level Explanations (NARRE) [2], and Dynamic Explainable Recommender (DER) [3]. Though these models achieve strong rating predictions, they are limited by the fact that they do not generate personalized reviews. The literature review continues by examining multi-task recommendation models such as Multi-Task Explainable Recommendation (MTER) [27] and Multimodal Review Generation (MRG) [25], which perform rating prediction and review text generation. Though these methods can perform well on both tasks, there are opportunities for improvement. One prominent limitation this paper seeks to address is that the multi-task models do not take advantage of the background information that is provided for each input-user pair. Due to its extensible architecture and relevance to the task at hand, MRG is selected as the model for improvement, and the proposed solution seeks to extend the model by incorporating a prototype editor. To the best of our knowledge, we are the first to consider prototype editing for the joint task of rating prediction and review text generation.

The proposed solution, MRG + Prototype Editor (MRG-PE) extends the MRG model so that it incorporates a prototype editing component. The prototype editor receives the previous reviews from the user and the previous reviews about the particular item, allowing extra input for each user-item pair. The prototype editor then outputs additional embeddings and passes them to a module for generating reviews. The solution is extended further with the model, MRG + Prototype Editor + Attention (MRG-PEA), which adds an attention mechanism to the prototype editor so that more

*All authors contributed equally to this research.

weight is given to the important words from the user and item reviews. The hypothesis is that the MRG-PEA model will outperform the original MRG model because the prototype editor will provide additional information about the item or user and the attention mechanism will filter out unimportant noisy information.

Experiments are run where the three models, MRG, MRG-PE, and MRG-PEA, are trained and tested on the Yelp dataset. The performance of the rating prediction is evaluated using metrics RMSE and MAE while the quality of the generated review text is assessed using metrics BLEU and ROUGE. An ablation analysis is then carried out, where the contributions of the prototype editor and the attention mechanism to the performance of the MRG model are studied. Results indicate that MRG-PEA outperforms MRG-PE, which in turn outperforms MRG, on the tasks of rating prediction and review text generation. Thus, we propose further research into the potential contributions of prototype editors with the attention mechanism in explainable recommender systems.

2 RELATED WORK

2.1 Explainable recommendation

2.1.1 Groups of existing methods. Explainable recommendation models consist of two main components, the model that generates the explanations and the display style of the explanation [30]. Popular explainable recommendation models can be grouped into the following categories:

- **Topic-based:** models that find hidden topics in the data.
- **Graph-based:** models that use a graph structure to represent relationships between users and items.
- **Matrix factorization (MF):** models that use matrix or tensor factorization to map user and item representations into a lower-dimensional space.
- **Deep learning:** models that use neural networks with many layers to extract high-level features in the data. Examples include Convolutional Neural Networks (CNN) for reading review text [23], attention mechanisms for putting weight on important words in review text [2], and Recurrent Neural Networks (RNN) to generate review texts [6].

Display styles for explanations can be grouped into the following categories:

- **User/item-based:** explanations that consider the interests of users that are similar to the user [21] or items that are similar to other items the user has shown interest in [22].
- **Feature-based:** explanations that compare features of the item (e.g. price, genre, tags) to demographic features of the user (e.g. age, gender, location) [31].
- **Textual:** explanations that extract aspects and sentiments from user-written texts to produce word clouds [28] or sentences. Sentences may be based on templates [29] or may be generated by Recurrent Neural Networks (RNN) [6].
- **Visual:** explanations that highlight regions of an image that are important for the recommendation [14].
- **Social:** explanations that consider the interests of people in the user's social or geographic network [18].

Due to the wide spread of solutions and the desire to use review data, this paper focuses on solutions using deep learning models and textual explanations. Within this group, methods can be distinguished based on whether they explain recommendations based on the content of the input or whether they explain recommendations using generated reviews.

2.1.2 Typical and recent methods. Many current recommender systems use deep learning techniques to learn from textual reviews. Deep Cooperative Neural Networks (DeepCoNN) use one network to learn user information from user reviews and one network to learn item information from item reviews, connecting them with a shared layer to learn hidden features [10, 32]. Experiments have demonstrated that DeepCoNN performs better than topic modeling systems, and that using synthetic reviews as training data for DeepCoNN leads to better recommendation ratings than human-written reviews [17]. Though these achievements highlight the potential of applying deep learning techniques in recommendation systems, there remains a limitation: no explanation is provided for the rating predictions.

In contrast, there are recommender systems that provide explanations according to the content of the reviews. Explicit Factor Model (EFM) is a matrix factorization model that "extracts product features and user opinions" by analyzing user reviews and then generates recommendations based on "product features" and latent features learned [29]. Many recommender systems now use deep learning techniques to extract explanatory information from textual reviews. A deep learning model that outperforms DeepCoNN and provides explainable predictions is Neural Attentional Regression model with Review-level Explanations (NARRE). Like DeepCoNN, NARRE uses a neural network to model the user and a neural network to model the item, but adds an attention mechanism to assign different weights to the reviews. As a result, NARRE can not only predict ratings, but also learn the usefulness of reviews and output the most useful reviews to explain the prediction [2]. A deep learning model that outperforms NARRE is the Dynamic Explainable Recommender (DER) model, which predicts ratings and highlights important phrases in reviews. DER uses a time-aware GRU to model dynamic user behaviour, which proves to be more effective than learning about the usefulness of reviews [3].

What these models lack, however, is a review generation component, which has proven to be a user-friendly method for explaining recommendations [30]. In recent years, multi-task recommendation models that jointly perform rating predictions and review text generation have gained traction. One of these models is the Multimodal Review Generation (MRG), which uses an MLP for rating prediction and an LSTM for review text generation, and shares user and item embeddings between both modules [25]. MRG outperforms DeepCoNN and MF models in rating prediction and LSTM-based models in review text generation, suggesting that a holistic representation, achieved by modeling both content and ratings, is advantageous. Multi-Task Explainable Recommendation (MTER) models user preference for recommendation and opinionated content for explanation using the joint tensor factorization [27]. Another solution that has been proposed in this category is a multi-task recommendation model that uses matrix factorization for rating predictions and adversarial sequence to sequence learning for recommendation explanations [15]. Multi-task methods are promising solutions for explainable recommendations but struggle

with the trade-off between accuracy and explainability. Their aim is to optimize multiple tasks and this can come at the expense of excelling at one task.

2.1.3 Differences with our model design. The proposed model, MRG-PEA, seeks to incorporate the strengths of existing methods as well as address the limitations outlined in the previous section. Since we build upon the multi-task model MRG, this means we automatically address some of the limitations of the content-based recommender systems. In comparison to DeepCoNN, the new model provides explanations for rating predictions, increasing user satisfaction. In comparison to EFM, NARRE, DER, the new model provides explanations in the form of generated natural language reviews that are tailored to the user and item. Though we interpret MRG as the most suitable model for extension, there remain areas of improvement. First, MRG does not take advantage of background information (i.e. other user and item reviews) that is available for each input-user pair. Second, MRG does not make use of a neural editor to improve the quality of the textual data it learns from. Thirdly, MRG does not use an attention mechanism to put more weight on important words such as powerful adjectives and less weight on unimportant parts of speech such as pronouns. The proposed model addresses these limitations by taking in additional reviews by the user and about the item, and using a prototype editor with an attention mechanism. These improvements are inspired by successful architectural details from existing methods, such as the Gated Recurrent Unit (GRU) in DER and the attention mechanism in NARRE.

2.2 Prototype editing

Prototype editing is a form of a generative language model that first samples a prototype sentence and then edits it into a new sentence. It has been shown that this increases the perplexity and generates higher quality outputs according to human evaluation [8].

2.2.1 Applications. Prototype editing has been used for tasks such as programming code auto-completion and the Hearthstone cards benchmark [9]. It can naturally be extended to all kinds of natural language tasks where an output text can be an edited version of an input text.

2.2.2 Relatedness to explainable recommendations. Text generation models typically involve LSTM models that generate text from scratch. [8] introduces a neural editor that samples a prototype from a training corpus and then edits it into a higher quality phrase. The neural editor makes use of an attention mechanism that places more weight on important words. There is evidence that attention mechanisms improve performance on sequence to sequence modeling tasks [26]. [9] builds upon the prototype editor approach by incorporating a learned retrieval pipeline.

3 METHOD

3.1 Multimodal Review Generation (MRG)

3.1.1 Motivation for extending MRG. After surveying various explainable recommendation methods, we conclude that the MRG model is a promising model to extend in order to obtain better performance. There are several motivations underlying this decision. To begin, our goal is to perform the joint task of predicting ratings and generating synthetic reviews for user-item pairs, with a special emphasis on the review generation part. Not only does the MRG model already perform both tasks, but it also outperforms several content-based recommendation systems, such as MF models, and review text generation models, such as LSTM models [25]. Moreover, the MRG model unifies the two tasks by sharing embeddings between the modules and combining the losses of the modules. Synergy is created between the modules, resulting in a holistic representation of the content and ratings that make it easier to generate relevant reviews for the user and item. Finally, the MRG has a simple, extensible architecture that is easy to build upon.

3.1.2 Model architecture. To begin with, we describe the original architecture of MRG [25]. The MRG model consists of a multi-layer

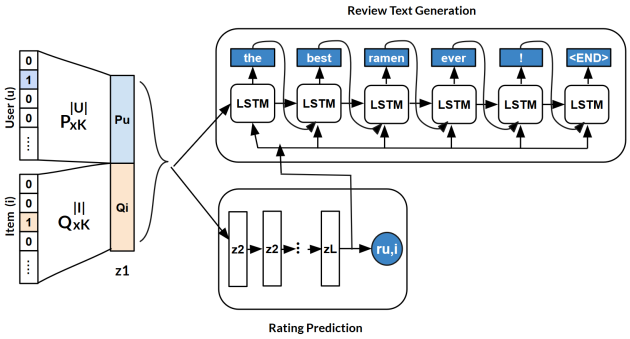

Figure 1: Architecture of MRG without image component.

perceptron (MLP) that performs rating prediction and an LSTM that performs review text generation. Additionally, there is a CNN that extracts visual features from images and passes annotated versions of the features to the model. In the scope of this paper, we disable the module with the CNN because we do not use data with images and the MRG model we refer to does not use the image component. An overview of the architecture is in Figure 1. The MRG model receives as input a user u and an item i, both of which are represented by one-hot vectors. The one-hot vectors are projected onto a user embedding P and an item embedding Q, to get specific embedding information for the user (P_u) and item (Q_i). The embedding vectors are concatenated into one vector z_1, which is passed as input to the MLP and used to initialize the LSTM embeddings. The vector z_1 is processed by the MLP in order to make a rating prediction $r_{u,i}$ and the final feature representation z_l from the MLP is passed to the LSTM. At each time step in the LSTM, z_l is concatenated with the embedding of the word generated by the previous time step and passed back to the LSTM. As a result, the output of the rating prediction module influences the review generation module at each time step. The input to the LSTM is a sequence of one-hot vectors, where each vector has the dimensionality of the vocabulary. The vectors are passed through pre-trained GloVe word embeddings [20], which are fine-tuned during training.

3.2 MRG + Prototype Editor (MRG-PE)

3.2.1 Model architecture. A Prototype Editor, in the form of a Gated Recurrent Unit encoder (GRU) [5], is added to the beginning of MRG to provide additional embeddings to the model.

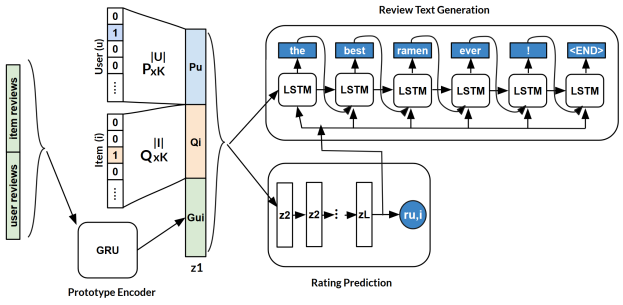

Figure 2: Architecture of MRG + Prototype Editor.

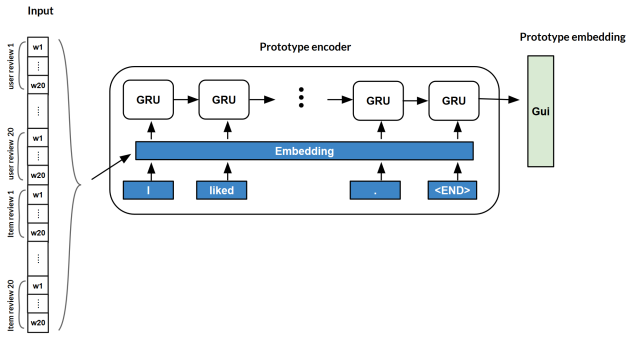

Figure 3: Visualization of Prototype Editor.

For each user-item pair that is passed into the model, 20 user reviews and 20 item reviews are retrieved. Each review is cut off after 20 words. In cases where there are not enough reviews or words, padding is used. The user reviews and item reviews are first concatenated into a 40×20 matrix, which is then reshaped into a vector with shape of 1×800. In this way, we can easily pass batches into the GRU during training and inference. The output of a GRU is used as an extra feature besides the user and item embedding, as shown in Figure 2. The prototype editor itself is shown in Figure 3 and the architecture of the GRU cell is shown in Figure 4.

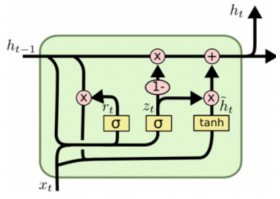

Figure 4: Visualization of Gated Recurrent Unit (GRU) [7].

3.2.2 Model learning. The model is trained in two parts using stochastic gradient descent with backpropagation. The rating prediction module is trained by minimizing the regularized squared error on the set of rating observations, where each observation is a triple {user, item, rating} [25]. The review generation module is trained by minimizing the regularized negative log-likelihood on the set of review observations, where each observation is a triple {user, item, review} [25]. Both losses are backpropagated to each relevant input, which includes the GloVe word embeddings, the prototype editor and the user and item embeddings. The losses of the modules are combined into a total loss function. During each batch, parameters for rating prediction are updated before parameters for review generation are updated.

3.3 MRG + Prototype Editor + Attention (MRG-PEA)

An attention mechanism is added to the prototype editor so that important words in the input sequence are given more weight and have a stronger influence on the prototype embedding. We use the attention mechanism from [4], which was originally an extension to the LSTM architecture but can also be applied to GRU cells in an RNN pipeline. Different from common encoder-decoder based attention (e.g. [1, 16]), [4] proposes a more general architecture that is suitable for networks with only an encoder. Specifically, for the update of hidden state at each time step, it explicitly computes a weighted average of the outputs from N previous states, where N is a hyperparameter known as the window size. Through this look-back mechanism, the network is able to learn a better representation of the input sequence.

4 EXPERIMENTS

4.1 Research questions

Our experiments are guided by the following research questions:

- **Question #1:** Does adding a prototype editor with an attention mechanism to MRG improve rating prediction?
- **Question #2:** Does adding a prototype editor with an attention mechanism to MRG improve review text generation?
- **Question #3:** How do the prototype editor and the attention mechanism contribute to the performance of MRG?

4.2 Experimental setup

4.2.1 Dataset. A subset of the Yelp dataset, as used in the experiments in [25], are experimented with. The full dataset consists of online reviews crawled from Yelp, covering 4 cities in the US: Chicago, Los Angeles, New York and San Francisco. Each online review consists of a rating, a review text, and one or more images taken by the user. In our experiments, only reviews from Chicago are used. We partition our data into a training set, a validation set and a test set respectively, where the validation data are used for selecting the best model during training.

4.2.2 Baselines. In addition to our proposed models, we reproduce the following baselines:

- NARRE [2]
- DER [3]
- DeepCoNN [32]

- MTER [27]
- MRG [25]

In most cases, the reported scores in the paper are similar to our own results, where sometimes the papers report better results and sometimes we outperform the papers. This is likely due to some stochasticity in the training process or due to different preprocessing of the data, as the datasets that were originally used by the authors are not always available. Apart from MRG, these results are not taken to be compared with the final models. DeepCoNN, DER and NARRE do not generate reviews and MTER was not trained on the YELP dataset. MTER does not generate full sentences but instead generates templates that are filled with personalized feature-opinion pairs.

4.2.3 Hyperparameter settings. We use the same hyperparameter settings for the MRG-based models (i.e. MRG, MRG-PE and MRG-PEA) as used in [25], which are shown in Table 1. However, for MRG-PEA, we adjust the epsilon value in the Adam optimizer to $1e-4$ to facilitate numerical stabilities. For all other baselines, the hyperparameter settings remain their original settings.

Table 1: Main hyperparameter settings of MRG, MRG-PE and MRG-PEA.

Parameter	Value
Learning rate	3e-4
Number of epochs	20
Batch size	64
Dropout probability	0.2
Dimensionality of latent factors	256
Maximum length of generated reviews	20
Dimensionality of word embeddings	200
Optimizer	Adam [12]

4.2.4 Evaluation metrics. To measure the performance of rating prediction, we use the metrics of Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE). To measure the semantic quality of the generated text, we use BLEU [19], which captures how much of the generated review overlaps with the human review, and ROUGE [13], which captures how much of the human review overlaps with the generated review. For BLEU, we look at n-grams from 1 to 4. For ROUGE, we look at n-grams 1 and 2 and the longest common sub-sequence (denoted as L).

4.3 Ablation Analysis

To examine the influence of individual components (i.e. Prototype Editor and Attention) on the overall performance of the MRG-PEA model, we perform an ablation analysis among MRG, MRG-PE and MRG-PEA. Table 2 and Table 3 show the performance of the three models evaluated on the test set in terms of rating prediction and review generation respectively. The rating prediction performance shows that adding a prototype editor to MRG decreases both MAE and RMSE, thus clearly improving the baseline. The introduction of the attention mechanism further decreases the RMSE, while it increases the MAE. The increase in MAE, while RMSE decreases, is likely due to the fact that RMSE penalizes larger errors more harshly. In other words, when adding the attention mechanism, the errors decrease in magnitude but increase in quantity. Therefore, the MAE is higher but the RMSE is lower. In addition, the review text generation performance shows that adding a prototype editor increases both BLEU and ROUGE scores of different lengths. The average increase in scores from MRG to MRG-PE is 9%, while with the attention mechanism added, the increase over MRG-PE is another 5%. This shows that the review generation pipeline benefits from incorporating the previous item reviews and user reviews using the prototype editor with attention mechanism.

Table 2: Ablation analysis: rating prediction performance (lower is better).

Model	MRG	✓	✓	✓
	Prototype Editor		✓	✓
	Attention			✓
Metric	MAE	0.783	**0.778**	0.793
	RMSE	1.019	1.008	**0.995**

Table 3: Ablation analysis: review text generation performance (higher is better). The ROUGE scores are reported based on F1-measures.

Model	MRG	✓	✓	✓
	Prototype Editor		✓	✓
	Attention			✓
Metric	BLEU-1	34.95	37.50	**44.29**
	BLEU-2	18.78	20.44	**21.62**
	BLEU-3	14.05	14.94	**15.55**
	BLEU-4	12.30	12.78	**13.07**
	ROUGE-1	22.38	22.89	**25.83**
	ROUGE-2	1.55	**2.02**	1.63
	ROUGE-L	16.44	17.14	**19.22**

5 DISCUSSION

After the analysis of our results, the previously introduced research questions can be answered. Including the prototype editor with attention mechanism in the existing MRG-model has shown to improve the output both in terms of rating prediction and review text generation. The prototype editor is especially important in predicting ratings, while the attention mechanism does not necessarily improve the results any further. For review text generation both components are important, and result in a 14% average increase of our evaluation metrics. It is therefore fair to assume that incorporating the previous item reviews and user reviews into the model increases the performance in both tasks. One should take into consideration, however, that this performance is based on the metrics used throughout the experiments. When looking at the actual generated texts, it becomes clear that methods that perform well on the evaluation metrics still have shortcomings. One of them, for example, is that some of the generated reviews are the same, even though their reference data {user, item, rating} and {user, item, review} differ. This behaviour makes it impossible for us to perform

qualitative case study and compare the different methods, as it is hard to trace back which reviews correspond to each other.

6 CONCLUSION

In this paper, different existing methods for explainable recommendation were studied. A new method, MRG-PEA, was proposed. This method builds upon existing state-of-the-art technique MRG with a prototype editor and an attention mechanism. Based on ablation analysis, in which the influence of individual components of the model was examined, it was demonstrated that the MRG-PEA model outperforms existing methods in the RMSE and most of the metrics that measure the semantic quality of the generated text. To improve upon this work, prototype editing could be incorporated into the decoding side of the model as well, using pointer networks. Another possible point of improvement is the enhancement of the loss function used in the MRG model. Currently, the review text generator is minimizing the regularized negative log-likelihood in order to update its parameters. Instead, one could use a Siamese LSTM network [24] to measure semantic similarity between a generated sentence and its reference sentence and use this as a signal for the loss. Finally, another possible direction of research could be in that of novel conditional transformer language model CTRL, which uses control codes to explicitly control the generated text [11]. This method is shown to have a promising performance on our task after feeding control codes related to the item. Unfortunately, it still lacks the computational efficiency due to the complexity of the model, making it difficult to fine-tune or even do inference.

REFERENCES

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural Machine Translation by Jointly Learning to Align and Translate. (2014). arXiv:cs.CL/1409.0473

[2] Chong Chen, Min Zhang, Yiqun Liu, and Shaoping Ma. 2018. Neural attentional rating regression with review-level explanations. In *Proceedings of the 2018 World Wide Web Conference*. International World Wide Web Conferences Steering Committee, 1583–1592.

[3] Yongfeng Zhang Chen, Xu and Zheng Qin. 2019. Dynamic Explainable Recommendation based on Neural Attentive Models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33.

[4] Jianpeng Cheng, Li Dong, and Mirella Lapata. 2016. Long short-term memory-networks for machine reading. *arXiv preprint arXiv:1601.06733* (2016).

[5] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. (2014). arXiv:cs.CL/1406.1078

[6] Felipe Costa, Sixun Ouyang, Peter Dolog, and Aonghus Lawlor. 2017. Automatic Generation of Natural Language Explanations. *CoRR* abs/1707.01561 (2017). arXiv:1707.01561 http://arxiv.org/abs/1707.01561

[7] Georgios Drakos. 2019. What is a Recurrent Neural Networks (RNNS) and Gated Recurrent Unit (GRUS). (February 2019). https://towardsdatascience.com/what-is-a-recurrent-nns-and-gated-recurrent-unit-grus-ea71d2a05a69

[8] Kelvin Guu, Tatsunori B. Hashimoto, Yonatan Oren, and Percy Liang. 2017. Generating Sentences by Editing Prototypes. *CoRR* abs/1709.08878 (2017). arXiv:1709.08878 http://arxiv.org/abs/1709.08878

[9] Tatsunori B Hashimoto, Kelvin Guu, Yonatan Oren, and Percy S Liang. 2018. A retrieve-and-edit framework for predicting structured outputs. In *Advances in Neural Information Processing Systems*. 10052–10062.

[10] Rose Catherine Kanjirathinkal. 2018. *Explainable Recommendations*. Ph.D. Dissertation. Stanford University.

[11] Nitish Shirish Keskar, Bryan McCann, Lav R Varshney, Caiming Xiong, and Richard Socher. 2019. Ctrl: A conditional transformer language model for controllable generation. *arXiv preprint arXiv:1909.05858* (2019).

[12] Diederik P. Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. (2014). arXiv:cs.LG/1412.6980

[13] Chin-Yew Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*. Association for Computational Linguistics, Barcelona, Spain, 74–81. https://www.aclweb.org/anthology/W04-1013

[14] Yujie Lin, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Jun Ma, and Maarten de Rijke. 2018. Explainable Fashion Recommendation with Joint Outfit Matching and Comment Generation. *CoRR* abs/1806.08977 (2018). arXiv:1806.08977 http://arxiv.org/abs/1806.08977

[15] Ruihai Dong Lu, Yichao and Barry Smyth. 2018. Why I like it: multi-task learning for recommendation and explanations. In *Proceedings of the 12th ACM Conference on Recommender Systems*. ACM.

[16] Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective Approaches to Attention-based Neural Machine Translation. (2015). arXiv:cs.CL/1508.04025

[17] Sixun Ouyang, Aonghus Lawlor, Felipe Costa, and Peter Dolog. 2018. Improving Explainable Recommendations with Synthetic Reviews. *CoRR* abs/1807.06978 (2018). arXiv:1807.06978 http://arxiv.org/abs/1807.06978

[18] Alexis Papadimitriou, Panagiotis Symeonidis, and Yannis Manolopoulos. 2012. A Generalized Taxonomy of Explanations Styles for Traditional and Social Recommender Systems. *Data Min. Knowl. Discov.* 24, 3 (May 2012), 555–583. https://doi.org/10.1007/s10618-011-0215-0

[19] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Philadelphia, Pennsylvania, USA, 311–318. https://doi.org/10.3115/1073083.1073135

[20] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 1532–1543.

[21] Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. 1994. GroupLens: An Open Architecture for Collaborative Filtering of Netnews. In *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work (CSCW '94)*. ACM, New York, NY, USA, 175–186. https://doi.org/10.1145/192844.192905

[22] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-based Collaborative Filtering Recommendation Algorithms. In *Proceedings of the 10th International Conference on World Wide Web (WWW '01)*. ACM, New York, NY, USA, 285–295. https://doi.org/10.1145/371920.372071

[23] Sungyong Seo, Jing Huang, Hao Yang, and Yan Liu. 2017. Interpretable Convolutional Neural Networks with Dual Local and Global Attention for Review Rating Prediction. 297–305. https://doi.org/10.1145/3109859.3109890

[24] Aditya Thyagarajan. 2015. Siamese Recurrent Architectures for Learning Sentence Similarity.

[25] Quoc-Tuan Truong and Hady Lauw. 2019. Multimodal Review Generation for Recommender Systems. In *The World Wide Web Conference (WWW '19)*. ACM, New York, NY, USA, 1864–1874. https://doi.org/10.1145/3308558.3313463

[26] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. *CoRR* abs/1706.03762 (2017). arXiv:1706.03762 http://arxiv.org/abs/1706.03762

[27] Nan Wang, Hongning Wang, Yiling Jia, and Yue Yin. 2018. Explainable Recommendation via Multi-Task Learning in Opinionated Text Data. *CoRR* abs/1806.03568 (2018). arXiv:1806.03568 http://arxiv.org/abs/1806.03568

[28] Yao Wu and Martin Ester. 2015. FLAME: A Probabilistic Model Combining Aspect Based Opinion Mining and Collaborative Filtering. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining (WSDM '15)*. ACM, New York, NY, USA, 199–208. https://doi.org/10.1145/2684822.2685291

[29] et al. Zhang, Yongfeng. 2014. Explicit factor models for explainable recommendation based on phrase-level sentiment analysis. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*. ACM.

[30] Yongfeng Zhang and Xu Chen. 2018. Explainable recommendation: A survey and new perspectives. *arXiv preprint arXiv:1804.11192* (2018).

[31] Xin Wayne Zhao, Yanwei Guo, Yulan He, Han Jiang, Yuexin Wu, and Xiaoming Li. 2014. We Know What You Want to Buy: A Demographic-based System for Product Recommendation on Microblogs. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '14)*. ACM, New York, NY, USA, 1935–1944. https://doi.org/10.1145/2623330.2623351

[32] Vahid Noroozi Zheng, Lei and Philip S. Yu. 2017. Joint deep modeling of users and items using reviews for recommendation. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*.